UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/966,615	09/28/2001	Daniel J. Strom	EMC01-09(01044)	4784

7590 10/01/2004

Barry W. Chapin, Esq.
CHAPIN & HUANG, L.L.C.
Westborough Office Park
1700 West Park Drive
Westborough, MA 01581

EXAMINER	
NGUYEN, VAN H	

ART UNIT	PAPER NUMBER
2126	3

DATE MAILED: 10/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/966,615	**Applicant(s)** STROM ET AL.	
	Examiner VAN H NGUYEN	**Art Unit** 2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 September 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-46* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *12/05/01*.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-46 are presented for examination.

Claim Objections

2. Claims 12, 31, 39, and 42 are objected to because of the following informalities:

"an second application programming interface definition" (claim 12, line 5; claim 31, line 12; claim 39, line 11; and claim 42, line 8) should read "a second application programming interface definition".

Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made

> This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 1-3, 5-10, 20-22, 24-29, 40, 41, 43, 45, and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Combs et al.** (U.S. 6,766,348).

5. As to claim 1, Combs teaches the invention substantially as claimed including a method for accessing a second application programming interface in a second computing environment from a first process operating in a first computing environment that cannot natively access the second application programming interface (col.18, lines 21-32), the method comprising the steps of:

detecting a first function call made by a first process in the first computing environment (col.4, lines 16-25);

based on the first function call, generating an encapsulated function call for transfer from the first computing environment to the second computing environment (col.18, lines 34-37); and

transferring the encapsulated function call containing the first parameter values from the first computing environment to the second computing environment (col.18, lines 37-39).

Combs does teach the encapsulated function call containing a mapping of first function call parameter values (col.10, lines 15-16) useable in the first computing environment to first parameter values (col.10, lines 22-39), but does not specifically teach meta parameter values.

It would have been obvious to one of ordinary skill in the art to have applied the teachings of Combs to include meta parameter values because it would have provided the capability for facilitating the transfer of function calls and data from application programs running on the local computer to the remote computer.

The fact that Combs' teachings "Arguments supplied to Bind include: user credentials; service priority; version; and the user session context" (col.10, lines 22-24) and purpose of providing user credentials, service priority, version, and the user session context arguments suggests meta parameter values.

5. As to claim 2, Combs teaches receiving an encapsulated response from the second computing environment, the encapsulated response containing second parameter values produced in the second computing environment from performance of at least one second function call in the second application programming interface that corresponds to the first function call detected in the first computing environment (col.4, lines 25-28); parsing the encapsulated response to map the second parameter values back to the first function call parameters usable by the first process in the first computing environment (col.18, lines 41-48).

6. As to claim 3, Combs teaches accessing a first application programming interface definition defining first function calls useable by first processes operating in the first computing environment, the first function calls corresponding to at least one second function call defined in a second application programming interface useable by second processes in the second computing environment (col.7, lines 21-26).

7. As to claim 5, Combs teaches the step of automatically deriving the first application programming interface definition from an analysis of the second application programming interface definition (col.3, lines 4-9).

8. As to claim 6, Combs teaches extracting the first function call parameters from the first function call using the first application programming interface (col.5, lines 58-61); and for each of the first function call parameters: selecting a respective first application programming

interface definition data structure in the first application programming interface that can represent the first function parameter as a first parameter (col.18, lines 28-32); and copying the first function call parameter into the respective first application programming interface definition data structure to produce a first parameter that contains the value of the first function call parameter (col.18, lines 34-37).

9. As to claim 7, Combs teaches the first parameters contain strings that represent values of the first function call parameters (col.7, lines 57-67 and col.8, lines 30-32).

10. As to claim 8, Combs teaches parsing the encapsulated response to detect second parameters (col.12, lines 51-67); for each of the second parameters detected: selecting a respective first application programming interface definition data structure in the first application programming interface that can represent the second parameter as a first function call parameter (col.18, lines 28-32); and copying second parameter values into the respective first application programming interface definition data structure to produce a first function call parameter that contains the value of the second parameter (col.18, lines 34-37).

11. As to claim 9, Combs teaches at least one second parameter in the encapsulated response identifies a first function call definition within the first application programming interface definition (col.10, lines 13-24), the first function call definition indicating the respective first application programming interface definition data structures that can accept the copied second parameter values (col.10, lines 24-39).

12. As to claim 10, Combs teaches the second parameter values represent values of parameters processed by second function calls in the second application programming interface in the second computing environment (col.10, lines 34-36) and wherein the first application

programming interface definition data structures that accept the copied second parameter values are natively incompatible for use in the second computing environment (col.10, lines 40-45).

13. As to claim 41, note the rejection of claim 1 above. Claim 41 is the same as claim 1, except claim 41 is a computer program product claim and claim 1 is a method claim.

14. As to claim 40, the rejection of claim 1 above is incorporated herein in full. Additionally, Combs further teaches an interface (101, fig.1), a memory (603, fig.6), a processor (604, fig.6); and an interconnection mechanism coupling the interface, the processor and the memory (fig.2).

15. As to claim 20, the rejection of claim 1 above is incorporated herein in full. Additionally, Combs further teaches an interface (101, fig.1), a memory (603, fig.6), a processor (604, fig.6); and an interconnection mechanism coupling the interface, the processor and the memory (fig.2) wherein the memory is encoded with a first application programming interface string generator application (col.7, lines 21-26) and a first parser application (col.12, lines 51-67) that, when performed on the processor, produce a respective first application programming interface string generator process and first parser process that cause the computer system to access a second application programming interface in a second computing environment by a first process operating in the computer system (col.18, lines 28-44).

16. As to claims 21, 22, and 24-29, they include the same subject matter as in claims 2, 3, and 6-10 above, and are similarly rejected under the same rationale.

17. As to claim 43, the rejection of claim 1 above is incorporated herein in full. Additionally, Combs further teaches mapping the second function call parameters to an encapsulated response (col.18, lines 40-42); and transferring the encapsulated response to the first computing environment (col.18, lines 43-44).

18. As to claim 45, it includes the same subject matter as in claim 7 above, and is similarly rejected under the same rationale.

19. As to claim 46, Combs teaches the encapsulated response includes at least one second parameter that contains a mapping of second function call parameters output from invocation of the second function (col.10, lines 40-55).

20. Claims 4, 11, 12-19, 23, 30-39, 42, and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Combs et al.** (U.S. 6,766,348) in view of **Saulpaugh et al.** (U.S. 6,298,345).

21. As to claim 4, Combs teaches the second function calls in the second application programming interface are defined in a C-based format useable by C-based second processes (col.7, lines 5-7). Combs, however, does not explicitly teach a Java-based format.

Saulpaugh teach a Java-based format (col.2, lines 57-65).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Saulpaugh and Combs because Saulpaugh's teaching would have provided the capability for increasing the flexibility of Combs' system and allowing the application programs to read, write, and maintain easier.

22. As to claim 11, it includes the same subject matter as in claim 4 above, and is similarly rejected under the same rationale.

23. As to claim 12, Combs teaches the invention substantially as claimed including a method for automatically generating applications allowing operation of an application programming interface in a second computing environment from a first process in a first

computing environment, in which the first process is not natively compatible with the second computing environment (abstract and col.18, lines 21-32), the method comprising the steps of:

analyzing a second application programming interface definition associated with a second computing environment to discover second function definitions in the second application programming interface definition (col.7, lines 18-26 and col.12, lines 51-67) ;

based on the step of analyzing, generating, for each second function definition discovered in the second application programming interface definition (col.18, lines 28-32):

a first string generator (901, fig. 9) capable of receiving a first function call in the first computing environment from a first process and encapsulating the first function call into an encapsulated function call (col.18, lines 33-37);

a second parser capable of receiving the encapsulated function call and parsing the encapsulated function call to invoke a corresponding second function definition in the second application programming interface definition for operation within the second computing environment (col.18, lines 37-39 and 50-59);

a second string generator (903, fig.9) capable of receiving an output from the second function call from a second process in the second computing environment and encapsulating the output into an encapsulated response (col.18, lines 40-42); and

a first parser capable of receiving the encapsulated response and parsing the encapsulated response to return the output to the first function in the first process operating in the first computing environment (col.18, lines 40-44).

Combs does teach generating the first function call and the encapsulated function call, but does not explicitly teach automatically generating.

Saulpaugh teach automatically generating (col.17, lines 19-26).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Saulpaugh and Combs because Saulpaugh's teaching would have provided the capability for facilitating the exchange of data between users of computational resources connected to a communications network and a distributed resource allocation system that manages the use of those computational resources.

24. As to claim 13, Combs teaches receiving a first application programming interface specific grammar produced as a result of the step of analyzing the second application programming interface definition (col.18, lines 28-32); and processing the first application programming interface specific grammar using a first application programming interface processor to produce the first parser by converting first function call definitions in the first application programming interface specific grammar into parser routines that can accept (col.18, lines 33-37) and parse parameters within encapsulated responses to provide first function call parameters back to first function calls associated with the first process that can operate in the first computing environment (col.18, lines 40-44).

25. As to claim 14, Combs teaches receiving a second application programming interface specific grammar produced as a result of the step of analyzing the second application programming interface definition (col.18, lines 28-32); and processing the second application programming interface specific grammar using a grammar processor to produce the second parser by converting second function call definitions in the second application programming interface specific grammar into parser routines that can accept (col.18, lines 39-41) and parse parameters within encapsulated function calls to provide second function call parameters to

second function calls associated with the second process that can operate in the second computing environment (col.18, lines 40-44).

26. As to claim 15, Combs teaches for each second function call definition discovered in the second application programming interface definition: producing a set of first application programming interface definition data structures that define first function call parameters that correspond to second function call parameters associated with the second function call definition (col.col.4, lines 16-20 and col.10, lines 13-16); and producing a set of first parameters that can represent the first application programming interface definition data structures (col.10, lines 22-39); and producing a first string generator function that can receive, from a first process that can operate in the first computing environment, a first function call that corresponds to the second function call definition (col.18, lines 28-37), the first string generator function capable of mapping the first function call parameters provided by the first function call to respective first parameters for encapsulation within an encapsulated function call for transfer to the second computing environment (col.18, lines 37-39).

27. As to claim 16, Combs teaches for each second function call definition discovered in the second application programming interface definition: producing a set of second parameters that can represent second function call parameters used by the second function call definition by accessing second application programming interface definition data structures defined in the second application programming interface definition (col.10, lines 22-39); and producing a second string generator function that can receive, from a second process that can operate in the second computing environment (col.18, lines 39-40), second function call parameters produced as output from the second process performing the second function call (col.18, lines 40-42), the

second string generator function capable of mapping the second function call parameters provided from invocation of the second function call to second parameters for encapsulation within an encapsulated response for transfer to the first computing environment (col.18, lines 40-44).

28. As to claim 17, Combs teaches analyzing the second application programming interface definition associated with the second computing environment to discover second data definitions in the second application programming interface definition (col.7, lines 18-26 and col.12, lines 51-67); and generating, for each second data definition discovered in the application programming interface definition (col.18, lines 28-32), a first application programming interface definition data structure capable of representing that second data definition in the second computing environment (col.18, lines 34-37).

29. As to claim 18, it includes the same subject matter as in claim 4 above, and is similarly rejected under the same rationale.

30. As to claim 19, Combs teaches the second application programming interface defines a set of second function definitions that provide access, via a second process, to data in a data storage system resource (col.4, lines 16-20).

31. As to claims 23 and 30, they include the same subject matter as in claim 4 above, and are similarly rejected under the same rationale.

32. As to claim 42, note the rejection of claim 12 above. Claim 42 is the same as claim 12, except claim 42 is a computer program product claim and claim 12 is a method claim.

33. As to claim 39, the rejection of claim 12 above is incorporated herein in full. Additionally, Combs further teaches a memory (603, fig.6), a processor (604, fig.6); and an interconnection mechanism (fig.6).

34. As to claim 31, the rejection of claim 12 above is incorporated herein in full. Additionally, Combs further teaches a memory (603, fig.6), a processor (604, fig.6), an interconnection mechanism (fig.6), and a grammar application programming interface processor application (col.17, lines 60-65).

35. As to claims 32-38, they include the same subject matter as in claims 13-19 above, and are similarly rejected under the same rationale.

Conclusion

36. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Stone (U.S. 6523137) teaches "Method and system for remote testing of application program interfaces."

- Allen et al. (U.S. 6502236) teaches "Method and apparatus for automatic generation of programs for processing data units of multiple formats."

- Miura et al. (U.S. 6334210) teaches "Language processing system and language processing method enabling reduction of memory region and overhead in profile information collection of computer."

- Hale et al. (U.S. 6151638) teaches "System and method for performing external procedure calls from a client program to a server program to a server program and back to the client program while both are running in a heterogeneous computer."

- De Bonet (U.S. 5905894) teaches "Meta-programming methods and apparatus."

- Priven et al. (U.S. 5327559) teaches "Remote and batch processing in an object oriented programming system."

37. Any inquiry concerning this communication or earlier communications from the examiner should be directed to VAN H. NGUYEN whose telephone number is (703) 306-5971. **After mid-October, 2004, the examiner can be reached at (571) 272-3765.** The examiner can normally be reached on Monday-Thursday from 8:30AM - 6:00PM. The examiner can also be reached on alternative Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (703) 305-9678.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VHN

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100